

CLAIM AMENDMENTS

Presented below is a complete listing of the claims of the present application indicating any amendments thereto.

1. (Original) A method to dynamically remove at least one selected module in a streaming data path of a graph having a plurality of modules, each module being connected to at least one other module to form the streaming data path, the streaming data path having at least one input module located at an input edge and at least one output module located at an output edge, the method comprising the steps of:

- sending a notification packet through the streaming data path to each module within the streaming data path, the notification packet indicating that data flow has stopped;

- detecting when the notification packet is received at each output module;

- commanding each selected module to be removed to change to a stop state after detecting when the notification packet is received at each output module;

- removing each selected module; and

- restarting data flow in the streaming data path.

2. (Original) The method of claim 1 further comprising the step of acquiring a graph lock.

3. (Original) The method of claim 2 further comprising the step of executing a multiple wait, the multiple wait specifying that it exits if at least one of the graph lock and an event type object is set.

4. (Original) The method of claim 1 further comprising the steps of:

- adding at least one additional module to the streaming data path after detecting when the notification packet is received at each output module; and

- commanding the additional module to change to a run state.

5. (Original) The method of claim 4 wherein each additional module has at least one pin, the step of adding at least one additional module comprises:

connecting each pin of the additional module to a pin of the module to which it is to be connected.

6. (Original) The method of claim 4 wherein each module has at least one pin, the method further comprising the steps of:

detecting when the input module receives the notification packet;

connecting at least one output pin of the input module to at least one input pin of the additional module; and

wherein the step of commanding each additional module to change to a run state is performed after the step of connecting the output pin of the input module to the input pin of the additional module.

7. (Original) The method of claim 1 wherein each module has at least one pin, the step of removing each selected module further comprises disconnecting each pin that is connected to the selected module prior to the step of removing each selected module.

8. (Original) The method of claim 1 further comprising the step of moving each selected module into a filter graph cache.

9. (Original) The method of claim 1 wherein each module has at least one pin, and at least two modules have at least one interface to support dynamic reconfiguration, one of the two modules being upstream of the selected module and the other of the two modules being downstream of the selected module, the method further comprising the steps of:

locating at least one input edge module, the input edge module being one of the two modules that is upstream of the selected module

locating at least one output edge module, the output edge module being the other of the two modules that is downstream of the selected module;

if there exists a first module other than the selected module between the input edge module and the output edge module:

- commanding the first module to change to a stop state;
- disconnecting each pin of the first module connected to the selected module;
- reconnecting each pin of the first module to a pin of an other module that was connected to the selected module; and
- commanding the first module to change to a run state.

10. (Original) The method of claim 9 further comprising the steps of
adding at least one additional module to the at least one streaming path; and
commanding the at least one additional module to change to a run state.
11. (Original) The method of claim 9 further comprising the steps of:
detecting when each input edge module receives a notification packet;
connecting at least one output pin of each input edge module to at least one input pin of the first module; and
wherein each first module is commanded to change to a run state when its input pin is connected to one of the first module and the input edge module;
12. (Original) The method of claim 9 further comprising the step of acquiring a graph lock.
13. (Original) A computer-readable medium having computer executable instructions for performing the steps recited in claim 1.
14. (Original) The computer-readable medium of claim 13 having computer executable instructions for performing the steps recited in claim 9.

15. (Original) A method to dynamically add at least one first module in a streaming data path of a graph having a plurality of modules, each module being connected to at least one other module to form the streaming data path, the streaming data path having at least one input module located at an input edge and at least one output module located at an output edge, the method comprising:

- sending a notification packet through the streaming data path to each module within the streaming data path, the notification packet indicating that data flow has stopped;
- detecting when the notification packet is received at each output module;
- adding each first module after detecting when the notification packet is received at each output module;
- commanding each first module to change to a run state; and
- restarting data flow in the streaming data path.

16. (Original) The method of claim 15 further comprising the step of acquiring a graph lock.

17. (Currently Amended) The method of claim 16 further comprising the step of executing a multiple wait, the multiple wait specifying that ~~the~~ it exits if one of the graph lock and an event type object is set.

18. (Original) The method of claim 15 further comprising the step of:

- removing at least one selected module from the streaming data path, the step of removing at least one selected module comprises:
 - commanding each of the selected module to be removed to change to a stop state; and
 - removing each selected module.

19. (Original) The method of claim 15 wherein each module has at least one pin, the step of adding each first module comprises:

for each pin of a module to be connected to the first module:

disconnecting the pin from each module it is connected to; and

connecting the pin to a pin of the first module.

20. (Original) The method of claim 15 wherein each module has at least one pin, the method further comprising the steps of:

detecting when the input module receives the notification packet;

connecting at least one output pin of the input module to at least one input pin of the first module; and

wherein the step of commanding each first module to change to a run state is performed after the step of connecting the input pin of the first module to at least one module.

21. (Original) The method of claim 15 wherein each module has at least one pin, at least two modules have at least one interface to support dynamic reconfiguration, one of the two modules being upstream of the first module and the other of the two modules being downstream of the first module, the method further comprising the steps of:

locating at least one input edge module, the input edge module being one of the at least two modules that is upstream of the first module;

locating at least one output edge module, the output edge module being the other of the two modules that is downstream of the first module;

if there exists a second module other than the first module between the input edge module and the output edge module:

commanding the second module to change to a stop state;

disconnecting each pin of the second module that is being connected to a pin of the first module and reconnecting it to the pin of the first module; and

commanding the second module to change to a run state.

22. (Original) The method of claim 21 further comprising the step of

removing at least one selected module to be removed from the at least one streaming path, the step of removing the selected module comprises the steps of:
commanding the selected module to change to a stop state;
disconnecting each pin that is connected to the selected module prior to removing the selected module; and
connecting each pin that was connected to the selected module to a pin of an other module that was connected to the selected module.

23. (Original) The method of claim 21 further comprising the steps of:
detecting when each input edge module receives a notification packet;
connecting at least one output pin of each input edge module to at least one input pin of one of the second module; and
wherein each second module is commanded to change to a run state when its input pin is connected to one of the second module and the input edge module.
24. (Original) The method of claim 21 further comprising the step of acquiring a graph lock.
25. (Original) A computer-readable medium having computer executable instructions for performing the steps recited in claim 15.
26. (Original) The computer-readable medium of claim 25 having further computer executable instructions for performing the steps recited in claim 21.
27. (Withdrawn) An interface for enabling dynamic changing of a streaming data path having a plurality of processing modules comprising:
a first command to determine if an input pin of a processing module can accept a media type on a next data sample;

a second command to provide notice when the processing module has processed data; and

a third command to signal when a reconnection should end at the input pin.

28. (Withdrawn) An interface for enabling dynamic changing of a streaming data path having a plurality of processing modules comprising a command to temporarily block data flow from an output pin of a processing module.

29. (Withdrawn) An interface for enabling dynamic changing of a streaming data path having a plurality of processing modules comprising:

a first command to perform a dynamic reconnection between an output pin and an input pin;

a second command to put a module into a cache;

a third command to remove a module from the cache;

a fourth command to enumerate modules in the cache;

a fifth command to get a start time used when a graph run call was last commanded; and

a sixth command to push data to a specified pin.

30. (Previously Presented) The method of claim 1 wherein each module provides an interface for enabling dynamic removing of the at least one selected module, the interface comprising:

a first command to determine if an input pin of a processing module can accept a media type on a next data sample;

a second command to provide notice when the processing module has processed data; and

a third command to signal when a reconnection should end at the input pin.

31. (Previously Presented) The method of claim 1 wherein each module provides an interface for enabling dynamic removing of the at least one selected module, the interface

comprising a command to temporarily block data flow from an output pin of a processing module.

32. (Previously Presented) The method of claim 1 wherein each module provides an interface for enabling dynamic removing of the at least one selected module, the interface comprising:

- a first command to perform a dynamic reconnection between an output pin and an input pin;
- a second command to put a module into a cache;
- a third command to remove a module from the cache;
- a fourth command to enumerate modules in the cache;
- a fifth command to get a start time used when a graph run call was last commanded; and
- a sixth command to push data to a specified pin.

33. (Previously Presented) The method of claim 15 wherein each module provides an interface for enabling dynamically adding the at least one first module, the interface comprising:

- a first command to determine if an input pin of a processing module can accept a media type on a next data sample;
- a second command to provide notice when the processing module has processed data; and
- a third command to signal when a reconnection should end at the input pin.

34. (Previously Presented) The method of claim 15 wherein each module provides an interface for enabling dynamically adding the at least one first module, the interface comprising a command to temporarily block data flow from an output pin of a processing module.

35. (Previously Presented) The method of claim 15 wherein each module provides an interface for enabling dynamically adding the at least one first module, the interface comprising:

- a first command to perform a dynamic reconnection between an output pin and an input pin;

- a second command to put a module into a cache;

- a third command to remove a module from the cache;

- a fourth command to enumerate modules in the cache;

- a fifth command to get a start time used when a graph run call was last commanded; and

- a sixth command to push data to a specified pin.